**ACKNOWLEDGEMENT**

A formal statement of thanks will hardly meet the ends of justice in the matter of expression of my deep sense of gratitude and obligation to all those who helped me in completion of this work.

A feeling of elation insists me in expressing my thanks to MD of EME Technologies for being my guide and providing me an insight of Android. This is his sincere support and consistent guidance that led to the completion of the project. His guidance gave immense confidence and encouragement that helped me to put in my best. I am deeply indebted to her as my project guide and providing constant and valuable guidance and making the project fruitful.

I would like to say special thanks to my friends for guiding me along the way towards completion of this project. In the end only this much can be said:

Student Names: -

Xxxxxx

**COMPANY PROFILE**

* 1. **COMPANY PROFILE**

EME Technologies is one of the best company in Chandigarh for Industrial Training since 2007 and most trustable company for web designing and development because it **believes in quality** and it delivers the best Quality to our clients. EME Technologies plans, designs, builds, tests and implements applications within strict quality standards, fully integrating them into clients' business processes and system environments. Their dedicated team has a unique combination of technical expertise, functional knowledge base, result oriented management and extensive experience.

They believe in developing long lasting business solution rather than just developing an application, which satisfies your needs for short period. Web Applications & Software application development at EME Technologies are robust, scalable and secure. Their ideas are flexible, scalable, secure and custom defined based on specific individual needs of our customers. With this they ensure that they follow the right strategy to ensure business renovation, lower operational costs and quick time to market. They ensure 100% success for their customers business. By improving reliability, speed and agility, they enable their customers to achieve sustainable advantage over their competitors.

## WEB DEVELOPMENT

Company offers complete business solutions including custom application development,  
web content management, managed web services, strategic planning and technologies consulting. Success of your online business depends on the website of your business which speaks the face of your organization. Since the primary goal of the website is to  
draw the targeted Visitors those who are interested for your services and products to increase your traffics, It must be developed to be informative, innovative and attractive.  
Serving to a diverse range of industries encompassing Dance and Culture, Tourism and Travels, Multi Level Marketing , E-commerce, Consulting, Hotel, Medical and Hospitals, E-governance, Corporate interactive informative website and many more, our website design and development solutions help you benefit from being on web.

## SOFTWARE DEVELOPMENT

They are specialized in the Development of custom software, Application Software, database.

Their team has spent many years searching for new ideas, studying Hundreds of different kinds of business and finding new effective Solutions.

* **MOBILE DEVELOPMENT**

## Android Application Development

EME Technologies is the fastest growing company in **Mobile application development** and now they have fortified their hands on **Android application development** to be also called as an **Android application development company / Agency** and they are moving continuously towards the path of success.

EME Technologies believes in providing customized services to all their clients which is accomplished by the team of skilled experts structuring desired Android Mobile application solutions. Their company works as a consultant when it comes to fine tuning the requirements and providing technical advices to make client's application a big success.

**Android** is an open source mobile operating system as well as software platform owned by Google. Its versatile features and open source has made it very popular in smart phones. Android applications are written in Java, with a rich set of libraries. Android Software Development Kit enables developers to develop custom Android applications using JAVA Programming language (J2ME).

**DIRECTOR:-**Mr. Sunil Bhutani

Email id:-[Sunil@emetechnologies.com](mailto:Sunil@emetechnologies.com)

* 1. **INTRODUCTION ABOUT PROJECT**

Today is time of mobile phones. Everybody wants to access everything on the phone like internet, games, business related application and **android,** an Operating System, helps to provide everybody these kinds of features. Now a day, we can access anything with android based mobile phones. Today everybody is using android phones and by creating applications for these types of phones, we can help people a lot.

This android based project, Banking App, is another application of android. This project is mainly created for bank, which gives information about bank account like how many money do you have, it provide the feature to diposite , draw the money from account.

**BankApp** is an android based on bank .  this application made to provide all feature or base option to user to add money , draw money from the account.By this application client can check its balance in account. Send the money from one account to another by online. You can pay ur bill by this application also.

The front end of application is **java/android** and back end is **SQLite.**

**1.3 FUNCTIONS AND SCOPE**

* This application help tp easy send money to any account.
* This application also help to easy to check your balance.
* This application is easy to use.
* This application containing all option or provide all facility to client.

**Chapter 3: Feasibility Study**

An initial investigation culminates in a proposal that determines whether a alternative system is feasible. When approved, the proposal initiates a feasibility study that describes and evaluates candidate system and provides for the selection of the best system that meets performance requirements. To do feasibility study, we need to consider the economic, technical and behavioral factors in system development. Many feasibility studies are disillusioning for both users and analysts. First, the study often presupposes that when the feasibility document is being prepared, the analyst is in a position to evaluate solutions. Second, most studies tend to overlook the confusion inherent in system development. The feasibility study is to serve as decision document; it must answer three key questions:

* Is there a new and better way to do the job that will benefit the user?
* What are the cost and savings of the alternatives?
* What is recommended?

A feasibility study is defined as an evaluation or analysis of the potential impact of a proposed project. A feasibility study is conducted to assist decision makers in or determining whether or not to implement a particular project program. It is performed to describe and evaluate candidate system and to provide for the selection of best system that meets system performance requirement.

During the course of completion of this project work the complete analysis of the proposed system was done. In the analyzing task a complete care about the feasibility of the proposed system was taken. The following feasibility analysis was carried during the course of this project work on “Customer relationship management”:

* Economic Feasibility
* Technical feasibility
* Operational feasibility
* Behavioral Feasibility
* Social Feasibility

**3.1 Economic Feasibility**

The cost involved in designing and implementation of the proposed system is as follows:

* **Analysis and Design Cost**: The cost of analysis & designing can be worked out by calculating the number of human days spend on the analysis and designing of the project and then multiplying the number of days with the cost of human day.
* **Programming Cost:** The cost is also calculated by calculating the number of human days spends on the coding of the project and then multiplying the number of the days with the cost of human day.
* **Stationary and Miscellaneous Expenses:** The cost of computer stationary is less than the cost of other not computer based stationary.

**3.2 Technical Feasibility**

The existing hardware and software facilities support the proposed system. Computers and storage media are available and the software can be developed. Basic technical requirement of the system and all aspects that the existing system facilities.

* **Hardware:** There’s need of Pentium machines with windows Server and printer for reports.
* **Manpower:** The technical and non-technical staff required to implementing this system.

**3.3 Operational Feasibility**

The present system is operationally feasible, as it has become easy to have details regarding which user has logged on the system and time of login, time of logon, and what information is transferred. The above details regarding the feasibility study show that the design of proposed system is very effective.

**3.4 Behavior Feasibility:**

This Website works in user-friendly environment. A person with the least knowledge of windows environment can use it effectively.

**3.5 Social Feasibility:**

Computers are known to facilitate change and people are usually restraint to change in determining the behavioral feasibility, we make an estimate of how strong reactions will the user staff makes towards the development of airline reservation system and try to keep the user response positive. Our proposed Website needs a user with the least knowledge of windows environment can use it effectively.

**Chapter 4: Hardware And Software Requirements**

For this project we need some special type of environment for setup. This is as follows:

4.1 .Minimum Hardware Requirements:

|  |  |  |
| --- | --- | --- |
| Sr.No. | Hardware | Specification |
| 1. | Android version | Kitket (4.4) |
| 2. | RAM | 1 GB |
| 3. | Internal Memory | 1 GB |
| 4. | CPU | 1.2GHz |

# **4.2 Minimum** Software Requirements:

|  |  |  |
| --- | --- | --- |
| Sr No. | Software | Specification |
|  | Operating System | Microsoft Windows 7 or above |
|  | Technology Used | Android and java. |
|  | Front-End | Android |
|  | IDE | Android studio |

**Chapter 5: Technology Used**

**5.1 Front End**

**5.1.1 Introduction to android:-**

[Operating Systems](http://www.engineersgarage.com/articles/operating-systems-tutorial) have developed a lot in last 15 years. Starting from black and white phones to recent smart phones or mini computers, mobile OS has come far away. Especially for smart phones, Mobile OS has greatly evolved from Palm OS in 1996 to Windows pocket PC in 2000 then to Blackberry OS and Android.

One of the most widely used mobile OS these days is ANDROID. Android is an operating system based on the Linux kernel, and designed primarily for touch screen mobile devices such as Smartphone and tablet computers. Android is a software bunch comprising not only operating system but also middleware and key applications. Android Inc. was founded in Palo Alto of California, U.S. by Andy Rubin, Rich miner, Nick sears and Chris White in 2003. Later Android Inc. was acquired by Google in 2005. The first publicly available smart phone running Android, the HTC Dream, was released on October 22, 2008.

The user interface of Android is based on direct manipulation , using touch inputs that loosely correspond to real-world actions, like swiping, tapping, pinching and reverse pinching to manipulate on-screen objects. Internal hardware such as accelerometers, gyroscopes and proximity sensors are used by some applications to respond to additional user actions, for example adjusting the screen from portrait to landscape depending on how the device is oriented. Android allows users to customize their home screens with shortcuts to applications and widgets, which allow users to display live content, such as emails and weather information, directly on the home screen. Applications can further send notifications to the user to inform them of relevant information, such as new emails and text messages.

Android's source code is released by Google under the Apache License, this permissive licensing allows the software to be freely modified and distributed by device manufacturers, wireless carriers and enthusiast developers. Most Android devices ship with a combination of open source and proprietary software. As of July 2013, Android has the largest number of applications available for download in Google Play store which has had over 1 million apps published, and over 50 billion downloads.

A developer survey conducted in April–May 2013 found that Android is the most used platform among developers: it is used by 71% of the mobile developers population. After original release there have been number of updates in the original version of Android.



**5.1.2 Android Versions**

If you've heard of Android, chances are you've heard all about its various versions. Some call it fragmentation, some say it's the nature of open-source, but in reality it's both a curse and a blessing. Regardless, it's good to have a little context about what all these version numbers and names mean when you see them posted on the Internet.

Each major version of Android has a dessert-based nickname, and they are all in alphabetical order. We like to think it's because of the delicious things they each have offered, but the folks at Google are pretty tight-lipped about why they used the internal code names they did. They certainly have a good sense of humour, and seem to like tasty deserts. Below is a quick primer on the different versions of Android that are still alive and kicking, from newest to oldest:

* **Android Marshmallow 6.0:-**

Android Marshmallow introduces a redesigned application permission model: there are now only eight permission categories, and applications are no longer automatically granted all of their specified permissions at installation time. An opt-in system is now used, in which users are prompted to grant or deny individual permissions (such as the ability to access the camera or microphone) to an application when they are needed for the first time. Applications remember the grants, which can be revoked by the user at any time. The new permission model is used only by applications compiled for Marshmallow using its [software development kit](https://en.wikipedia.org/wiki/Software_development_kit) (SDK), and older apps will continue to use the previous all-or-nothing permission model. Permissions can still be revoked for those apps, though this might prevent them from working properly, and a warning is displayed to that effect.

****

* **Android 5.0X:**

Android Lollipop is a version of the [Android](https://en.wikipedia.org/wiki/Android_%28operating_system%29) [mobile operating system](https://en.wikipedia.org/wiki/Mobile_operating_system) developed by [Google](https://en.wikipedia.org/wiki/Google), spanning versions between 5.0 and 5.1.1.[[3]](https://en.wikipedia.org/wiki/Android_Lollipop#cite_note-3) Unveiled on June 25, 2014, during the [Google I/O](https://en.wikipedia.org/wiki/Google_I/O) conference, it became available through official [over-the-air](https://en.wikipedia.org/wiki/Over-the-air_programming) (OTA) updates on November 12, 2014, for select devices that run distributions of Android serviced by Google (such as [Nexus](https://en.wikipedia.org/wiki/Google_Nexus) and [Google Play edition](https://en.wikipedia.org/wiki/Google_Play_edition) devices). Its source code was made available on November 3, 2014.One of the most prominent changes in the Lollipop release is a redesigned user interface built around a [design language](https://en.wikipedia.org/wiki/Design_language) known as "[Material design](https://en.wikipedia.org/wiki/Material_design)". Other changes include improvements to the notifications, which can be accessed from the lock screen and displayed within applications as top-of-the-screen banners. Google also made internal changes to the platform, with the [Android Runtime](https://en.wikipedia.org/wiki/Android_Runtime) (ART) officially replacing [Dalvik](https://en.wikipedia.org/wiki/Dalvik_virtual_machine) for improved application performance, and with changes intended to improve and optimize battery usage.



* **Android 4.4X:-**

The Android 4.4 KitKat operating system uses advanced memory optimization technologies. As a result, it is available on [Android](http://searchenterpriselinux.techtarget.com/definition/Android) devices with as little as 512 [MB](http://searchstorage.techtarget.com/definition/megabyte) of [RAM](http://searchmobilecomputing.techtarget.com/definition/RAM).  This is important because previous versions of the operating system required more internal memory which made them incompatible with many older device models. This was a major cause of the problem known as Android [fragmentation](http://searchconsumerization.techtarget.com/definition/mobile-device-fragmentation). KitKat also boasts an improved security-enhanced [Linux](http://searchenterpriselinux.techtarget.com/definition/Linux) [module](http://searchenterpriselinux.techtarget.com/definition/module) which helps prevent, unauthorized [app](http://searchconsumerization.techtarget.com/definition/app) access. Google announced Android 4.4 KitKat in September 2013 and debuted it on the [Nexus](http://searchconsumerization.techtarget.com/definition/Nexus) 5 smartphone the following month. For months prior to the release, observers expected the OS to be called Key Lime Pie, but Google renamed it as part of a marketing promotion with the Hershey Company, which sells a candy bar called Kit Kat in the United States.



* **Android 4.1-4.2 – Jelly Beans:-**

Jelly Bean arrived at Google IO 2012, with the release of the [ASUS Nexus 7](http://androidcentral.com/google-nexus-7), followed by a quick update for unlocked Galaxy Nexus phones. Later in the year, the release of the [Nexus 10](http://androidcentral.com/samsung-nexus-10) and [Nexus 4](http://androidcentral.com/lg-nexus-4) updated things from 4.1 to 4.2, but the version remained Jelly Bean. The release [polished the UI design](http://developer.android.com/guide/practices/ui_guidelines/index.html) started in Ice Cream Sandwich, and brought several great new features to the table.

Besides the new focus on responsiveness with [Project Butter](http://youtu.be/V5E5revikUU), Jelly Bean brings multi-user accounts, actionable notifications, lock screen widgets, quick-settings in the notification bar, Photosphere to the "stock" Android camera and Google Now.

Jelly Bean is hailed by many as the turning point for Android, where all the great services and customization options finally meet great design guidelines. It's certainly very visually pleasing, and we'd argue that it's become one of the nicest looking mobile operating systems available.



* **Android 4.0 - Ice Cream Sandwich:-**

The follow-up to Honeycomb was announced at [Google IO](http://www.androidcentral.com/tags/google-io) in May 2011 and released in December 2011. Dubbed Ice Cream Sandwich and finally designated Android 4.0, Ice Cream Sandwich brings many of the design elements of Honeycomb to smartphones, while refining the Honeycomb experience.

The first device to launch with ICS was the [Samsung Galaxy Nexus](http://www.androidcentral.com/samsung-galaxy-nexus). The Motorola Xoom and the ASUS Transformer Prime were the first tablets to receive updates, while the Samsung Nexus S was the first smartphone to make the jump to Android 4.0.



* **Android 3.X – Honeycomb:-**

Android 3.0 came out in February 2011 with the Motorola Xoom . It's the first version of Android specifically made for tablets, and brings a lot of new UI elements to the table.  Things like a new System bar at the bottom of the screen to replace the Status bar we see on phones, and a new recent applications button are a great addition for the screen real estate offered by Android tablets.

Some of the standard Google applications have also been updated for use with Honeycomb, including the Gmail app and the Talk app.  Both make great use of [fragments](http://www.androidcentral.com/fragments-explained-and-will-be-backwards-compatible-donut), and the Talk app has video chat and calling support built in.  Under the hood, 3D rendering and hardware acceleration have been greatly improved.

We can't talk about Honeycomb without mentioning that it also shows [Google's new distribution method](http://www.androidcentral.com/google-tightening-reins-android), where manufacturers are given the source code and license to use it only after their hardware choices have been approved by Google.  This dampens third party development, as the source code is no longer available for all to download and build, but Google assures us they will address this issue in the future.

Improvements to Honeycomb were announced at Google IO in May 2011 as Android 3.1, and Android 3.2 has followed.



* **Android 2.3 :-**

Android 2.3 came out of the oven in December 2010, and like Eclair, has a new "Googlephone" to go along with – the nexus S Gingerbread brings a few UI enhancements to Android, things like a more consistent feel across menus and dialogs, and a new black notification bar, but still looks and feels like the Android we're used to, with the addition of a slew of language support.

Gingerbread brings support for new technology as well.  [NFC (Near Field Communication)](http://androidcentral.com/google-and-nxp-integrate-nfc-android-23-starting-nexus-s) is now supported, and SIP (Internet calling) support is now native on Android. Further optimizations for better battery life round out a nice upgrade.

[Behind the scenes](http://androidcentral.com/gingerbread-under-hood), the fellows at Mountain View spent time with more JIT (the Just-In-Time compiler) optimizations, and made great improvements to Androids garbage collection, which should stop any stuttering and improve UI smoothness.  Round that out with new a multi-media framework for better support of sound and video files.



* **Android 2.2 – Froyo:-**

Android 2.2 was announced in May 2010 at the Google IO conference in San Francisco. The single largest change was the introduction of the Just-In-Time Compiler -- or JIT -- which [significantly speeds up the phone's processing power.](http://www.androidcentral.com/benchmarking-android-22-froyo-against-android-21-eclair) Along with the JIT, Android 2.2 also brings [support for Adobe Flash 10.1](http://www.androidcentral.com/froyo-feature-adobe-flash-101-beta). That means you can play your favorite Flash-based games in Android's web browser.



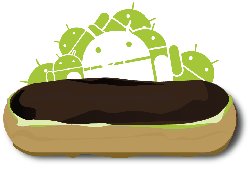
* **Android 2.0/2.01/2.1 – Éclair:-**

Eclair was a pretty major step up over its predecessors. Introduced in late 2009, Android 2.0 first appeared on the Motorola Droid, bringing improvements in the browser, Google Maps, and a new user interface. [Google Maps Navigation](http://www.androidcentral.com/goolge-introduces-turn-turn-directions-google-maps-navigation-android-20) also was born in Android 2.0, quickly bringing the platform on par with other stand-along GPS navigation systems.

Android 2.0 quickly gave way to 2.0.1, which the Droid received in December 2009, mainly bringing bugfixes. And to date, the Droid remains the phone phone to have explicitly received Android 2.0.1.

The [now-defunct Google Nexus One](http://www.androidcentral.com/google-nexus-one-no-longer-sale) was the first device to receive Android 2.1 when it launched in January 2010, bringing a souped-up UI with cool 3D-style graphics. From there, the rollout of Android 2.1 has been relatively slow and painful. Manufacturers skipped Android 2.0 in favor of the latest version but needed time to tweak their customizations, such as Motorola's Motoblur.

HTC's Desire and Legend phones launched with Android 2.1 later in the year, touting a new and improved Sense user interface.

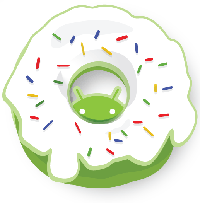


* **Android 1.6 – Donut:-**

Donut, released in September 2009, built on the features that came with Android 1.5, and expanded them. While not very rich in the eye-candy department, Android 1.6 made some major improvements behind the scenes, and provided the framework base for the amazing features to come.  To the end user, the two biggest changes would have to be the improvements to the [Android Market](http://www.androidcentral.com/search/market), and universal search.

Behind the screen, Donut brought support for higher resolution touchscreens,  much improved camera and gallery support, and perhaps most importantly, native support for Sprint and Verizon phones. Without the technology in Android 1.6, there would be no Motorola Droid X or HTC Evo 4G.

The devices released with Android 1.6 cover a wide range of taste and features, including the [Motorola Devour](http://www.androidcentral.com/search/devour), the [Garminphone](http://www.androidcentral.com/search/garminphone) and the [Sony Ericsson Xperia X10](http://www.androidcentral.com/search/x10).



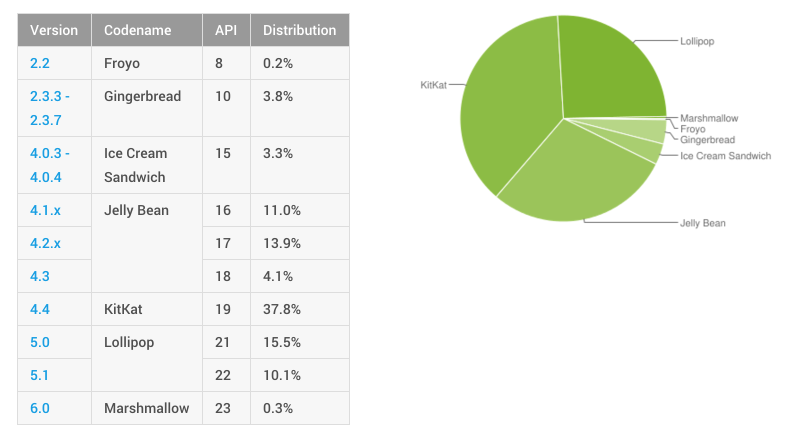
* **Android 1.5 – Cupcake:-**

Cupcake was the first major overhaul of the Android OS.  The Android 1.5 SDK was released in April 2009 and brought along plenty of UI changes, the biggest probably being support for and folders on the homescreens. There were plenty of changes behind the scenes, too.  Cupcake brought features like improved Bluetooth support, camcorder functions, and new upload services like YouTube and Picasa. Android 1.5 ushered in the era of the modern Android phone, and the explosion of devices included favorites like the HTCHero/Eris, the Samsung Moment ,and the Motorola cliq.



**5.1.3 Platform usage:-**

These charts provide data about the relative number of devices accessing the Play Store recently and running a given version of the Android platform, as of January 11, 2014.



**5.1.4 JAVA**

* **Key Features of the Java Language:-**The Java language has key features that make it ideal for developing server applications. These features include:
* **Simplicity**. Java is a simpler language to master than most others you use in server applications because of its consistent enforcement of the object model. The large standard set of class libraries brings powerful tools to Java developers on all platforms.
* **Portability**. Java is ubiquitous due to its portability across platforms. It is possible to write platform-dependent code in Java, but it is also simple to write programs that move seamlessly across machines. Server applications, which by their nature do not support graphical user interfaces directly on the platform that hosts them, also tend to avoid the few real platform portability issues that Java has.
* **Automatic Storage Management**. The Java virtual machine automatically performs all memory allocation and de-allocation during program execution. Java programmers can neither allocation nor free memory explicitly. Instead, they depend on the virtual machine to perform these bookkeeping operations, allocating memory as they create new objects and de-allocating memory when the objects are no longer referenced. The latter operation is known as garbage collection.

* **Strong Typing**. Before you use a Java variable, you must declare the class of the object it will hold. Java's strong typing makes it possible to provide a reasonable and safe solution to inter-language calls in the case of Java and PL/SQL and to integrate Java and SQL. Languages without strong typing are typically more flexible at development time but make it difficult to build safe programs because they cannot take advantage of type information at compile time.

* **No Pointers**. Although Java retains much of the flavor of C in its syntax, it does not support direct pointers or pointer manipulation. You pass all parameters except primitive types by reference (that is, object identity is preserved), not by value. Java does not provide C's low level; direct access to pointers, thereby eliminating the major source of bugs, memory corruption, and memory leaks that plague C programs and that in turn prevents Oracle from allowing such programs to execute in the same address space as the RDBMS.

* **Exception Handling**. Exception handling is a feature of the Java language. Java exceptions are objects. Exception handling is a powerful tool for developers to build robust programs. Java even goes as far as requiring developers to declare which exceptions may be thrown (or signaled) by methods in any particular class.

* **Flexible Namespace**. Java defines classes and holds them within a hierarchical structure that mirrors the Internet's domain namespace. This approach enables you to distribute Java applications while avoiding the possibility of name collisions. Java extensions such as the Java Naming and Directory Interface (JNDI) that Oracle uses, for example, in its EJB implementation even provide a framework for multiple name services to be federated. Java's namespace approach is flexible enough for Oracle to incorporate the concept of a schema in the way class names are resolved while fully complying with the language specification.

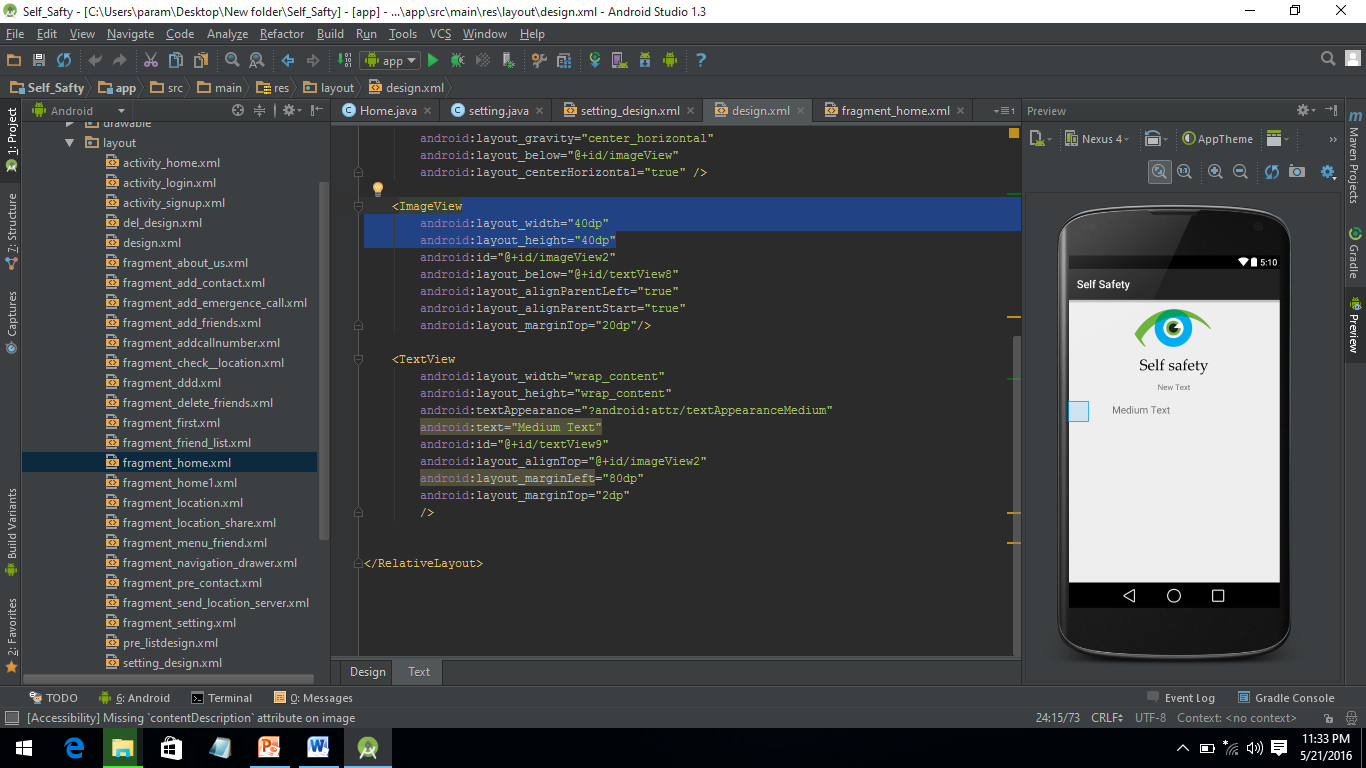
* **Security.** The design of Java byte codes and the virtual machine allow for built-in mechanisms to verify that Java binary code has not been tampered with in ways that would violate the safety and security restrictions the language supports. SQLITE is installed within instance of Security Manager that uses Oracle security to further enhance the features the language supports. The Security Manager uses Oracle roles to restrict access to sensitive functionality such as disk file manipulation while fully complying with the requirements of the JLS.

* **Standards for Connectivity to Relational Databases**. JDBC and SQL provide important and standard ways for programs you write in a general purpose object-oriented programming language-- Java--to access and manipulate data resident in relational databases. Vendors such as Oracle provide vendor-specific drivers that allow vendor-independent, portable Java code to access a specific relational database. Java developers can use vendor-specific extensions to access vendor-specific features.
* **Speed**. Because Java executes platform-independent byte codes on top of a virtual machine, which in turn deals with the specific hardware platform, a degree of inefficiency is inherent in Java byte code interpreters as compared to a language such as C. Virtual machine suppliers such as Oracle address this speed issue in different ways. One popular approach is to use a Just in Time (JIT) compiler. JITs quickly compile Java byte codes to native (platform-specific) machine code, allowing Java code that is run frequently to be executed at speeds closer to languages such as C. Oracle has adopted a Way Ahead of Time (WAT) approach to gain speed. In the WAT approach, Aurora translates Java byte codes to platform-independent C code, which a standard C compiler then compiles for the target platform.
* **Robust**. Java is designed to eliminate certain types of programming errors. Java is strongly typed, which allows extensive compile-time error checking. It does not support memory pointers, which eliminates the possibility of overwriting memory and corrupting data. In addition, its automatic memory management (garbage collection) eliminates memory leaks and other problems associated with dynamic memory allocation/de-allocation.

**5.1.5 Android Studio**

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on [IntelliJ IDEA](https://www.jetbrains.com/idea/). On top of IntelliJ's powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building Android apps, such as:

* A flexible Gradle-based build system
* A fast and feature-rich emulator
* A unified environment where you can develop for all Android devices
* Instant Run to push changes to your running app without building a new APK
* Code templates and GitHub integration to help you build common app features and import sample code
* Extensive testing tools and frameworks
* Lint tools to catch performance, usability, version compatibility, and other problems
* C++ and NDK support
* Built-in support for [Google Cloud Platform](http://developers.google.com/cloud/devtools/android_studio_templates/), making it easy to integrate Google Cloud Messaging and App Engine



**PROJECT STRUCTURE**

Each project in Android Studio contains one or more modules with source code files and resource files. Types of modules include:

* Android app modules
* Library modules
* Google App Engine modules

By default, Android Studio displays your project files in the Android project view, as shown in figure 1. This view is organized by modules to provide quick access to your project's key source files.

All the build files are visible at the top level under **Gradle Scripts** and each app module contains the following folders:

* **manifests**: Contains the AndroidManifest.xml file.
* **java**: Contains the Java source code files, including JUnit test code.
* **res**: Contains all non-code resources, such as XML layouts, UI strings, and bitmap images.

The Android project structure on disk differs from this flattened representation. To see the actual file structure of the project, select **Project** from the **Project** dropdown (in figure 1, it's showing as **Android**).

You can also customize the view of the project files to focus on specific aspects of your app development. For example, selecting the **Problems** view of your project displays links to the source files containing any recognized coding and syntax errors, such as a missing XML element closing tag in a layout file.

**User Interface**

* The **toolbar** lets you carry out a wide range of actions, including running your app and launching Android tools.
* The **navigation bar** helps you navigate through your project and open files for editing. It provides a more compact view of the structure visible in the Project tool window.
* The **editor window** is where you create and modify code. Depending on the current file type, this window can change. For example, when viewing a layout file, the editor window displays the layout editor and offers the option to view the corresponding XML file.
* **Tool windows** give you access to specific tasks like project management, search, version control, and more. You can expand them and collapse them.
* The **status bar** displays the status of your project and the IDE itself, as well as any warnings or messages.

You can organize the main window to give yourself more screen space by hiding or moving toolbars and tool windows. You can also use keyboard shortcuts to access most IDE features.

At any time, you can search across your source code, databases, actions, elements of the user interface, and so on, by double-pressing the Shift key, or clicking the magnifying glass in the upper right-hand corner of the Android Studio window. This can be very useful if, for example, you are trying to locate a particular IDE action that you have forgotten how to trigger.

**Window Tools**

Instead of using preset perspectives, Android Studio follows your context and automatically brings up relevant tool windows as you work. By default, the most commonly used tool windows are pinned to the tool window bar at the edges of the application window.

* To expand or collapse a tool window, click the tool’s name in the tool window bar. You can also drag, pin, unpin, attach, and detach tool windows.
* To return to the current default tool window layout, click **Window > Restore Default Layout** or customize your default layout by clicking **Window > Store Current Layout as Default**.
* To show or hide the entire tool window bar, click the window icon in the bottom left-hand corner of the Android Studio window.
* To locate a specific tool window, hover over the window icon and select the tool window from the menu.

You can also use keyboard shortcuts to open tool windows. Table 1 lists the shortcuts for the most common windows.

**Gradle Build System**

Android Studio uses Grasdle as the foundation of the build system, with more Android-specific capabilities provided by the [Android Plugin for Gradle](https://developer.android.com/tools/revisions/gradle-plugin.html). This build system runs as an integrated tool from the Android Studio menu, and independently from the command line. You can use the features of the build system to do the following:

* Customize, configure, and extend the build process.
* Create multiple APKs for your app, with different features using the same project and modules.
* Reuse code and resources across source sets.

By employing the flexibility of Gradle, you can achieve all of this without modifying your app's core source files. Android Studio build files are named build gradle. They are plain text files that use [Groovy](http://groovy.codehaus.org/) syntax to configure the build with elements provided by the Android plugin for Gradle. Each project has one top-level build file for the entire project and separate module-level build files for each module. When you import an existing project, Android Studio automatically generates the necessary build files.

**5.2 BackEnd:**

**5.2.1 SQLite**

SQLite is a software library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. SQLite is the most widely deployed SQL database engine in the world. The source code for SQLite is in the public domain. This tutorial will give you a quick start with SQLite and make you comfortable with SQLite programming.



SQLite is an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. It is a database, which is zero-configured, which means like other databases you do not need to configure it in your system.

SQLite engine is not a standalone process like other databases, you can link it statically or dynamically as per your requirement with your application. SQLite accesses its storage files directly.

keeping the same [API](https://en.wikipedia.org/wiki/Application_programming_interface) as mSQL. By keeping the API consistent with the mSQL system, many developers were able to use SQLite instead of the (proprietarily licensed) mSQL antecedent.

### **5.2.2 Self-Contained**

SQLite is self-contained means it requires minimal support from the operating system or external library. This makes SQLite usable in any environments especially in embedded devices like iPhones, Android phones, game consoles, handheld media players, etc.

SQLite is developed using ANSI-C. The source code is available as a big sqlite3.c and its header file sqlite3.h. If you want to develop an application that uses SQLite, you just need to drop these files into your project and compile it with your code.

### **Zero-configuration**

Because of the serverless architecture, you don’t need to “install” SQLite before using it. There is no server process that needs to be configured, started, and stopped.

In addition, SQLite does not use any configuration files.

### **Transactional**

All transactions in SQLite are fully ACID-compliant. It means all queries and changes are Atomic, Consistent, Isolated, and Durable.

In other words, all changes within a transaction take place completely or not at all even when an unexpected situation like application crash, power failure, or operating system crash occurs.

## **SQLite distinctive features**

SQLite uses dynamic types for tables. It means you can store any value in any column, regardless of the data type.

SQLite allows a single database connection to access multiple database files simultaneously. This brings many nice features like joining tables in different databases or copying data between databases in a single command.

SQLite is capable of creating in-memory databases which are very fast to work with.

**Chapter 6: Data Flow Diagram**

A **Data Flow Diagram** (**DFD**) is a graphical representation of the "flow" of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated. DFDs can also be used for the visualization of data (structured design).

Data flow diagrams are used to study information flow and information through a computer based system. DFD’s were introduced by De Marso (1978) and Gane and Sarson (1979) and is the important tool used by system analysis. The main advantage of DFD is that it can provide an overview of what data a system would process, what transformation of data are done, what data are stored and which stored data are used and where the result stores. The graphic illustration of a system makes it easy between a user and where the result stores. DFD’s are structured in such a way that starting forms a diagram, which gives a broad overview at a glance; they can be extended to a hierarchy of diagrams giving more and more detail. We have not found the consistency and redundancy at the time of normalization.

A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of process or information about whether processes will operate in sequence or in parallel

At its simplest, a data flow diagram looks at how data flows through a system. It concerns things like where the data will come from and go to as well as where it will be stored. But you won't find information about the processing timing (e.g. whether the processes happen in sequence or in parallel).

We usually begin withdrawing a **context diagram**, a simple representation of the whole system. To elaborate further from that, we drill down to a **level 1 diagram** with additional information about the major functions of the system. This could continue to evolve to become a level 2 diagram when further analysis is required. Progression to level 3, 4 and so on are possible but anything beyond level 3 is not very common. Please bear in mind that the level of detail asked for depends on your process change plan.

* 1. **:DFD Symbols: -**

A square defines a source or destination of system data.

An arrow identifies the data flow or data in motion. It is a pipeline through which information flows.

A circle or bubble represents a process transform incoming data flow in to outgoing data flow.

Horizontal line represents data stored or data at rest or a temporary rest repository of data.

**Chapter 7: ER- Diagram**

**Entity – Relationship Model:**

As a database designer one should use an Entity Relationship (ER) diagram as a tool to build the logical database design of a system. An ER model is typically implemented as a database. In the case of a relational database, which stores data in tables, every row of each table represents one instance of an entity. Some data fields in these tables point to indexes in other tables; such pointers represent the relationships. An ER diagram represents the following three elements**:**

**Entities:** An entity is an object with a distinct set of properties that is easily identified. Entities are the building blocks of a database. You represent an entity using a rectangular box that contains the name of the entity**.**

**Attributes**: An attribute is a property of an entity that differentiates it from other entities and provides information about the entity. An attribute type is a property of an entity type**.**

**Relationships:** A relationship is a crucial part of the design of a database. It is used to establish a connection between a pair of logically related entities. Separate entities can have relationship with each other. Relationship is represented between two entities using a diamond labeled with the name of the relationship.

Following ER Model:

Entity

Attribute

Multi valued Attribute

Relationship

Data flow

Database

**Chapter 9: Testing**

The development process involves various types of testing. Each test type addresses a specific testing requirement. Software testing is used in association with verification and validation:

**Verification**: Have we built the software right (i.e., does it match the specification)?

**Validation**: Have we built the right software (i.e., is this what the customer wants)?

First, testing is about verifying that what was specified is what was delivered: it verifies that the product meets the functional, performance, design, and implementation requirements identified in the procurement specifications.

Second, testing is about managing risk for the acquiring agency and the system’s vendor/ developer/ integrator. The testing program is used to identify when the work has been “completed” so that the contract can be closed, the vendor paid, and the system shifted by the agency into the warranty and maintenance phase of the project.

**Scope:-**

Software testing may be viewed as an important part of the software quality assurance (SQA) process.

In SQA, software process specialists and auditors take a broader view on software and its development. They examine and change the software engineering process itself to reduce the amount of faults that end up in defect rate. What constitutes an acceptable defect rate depends on the nature of the software although there are close links with SQA testing departments often exist independently, and there may be no SQA areas in some companies.

**Testing: -**

**1. Planning**

This involves writing and reviewing unit, integration, functional, validation and acceptance

test plans. Planning in testing of Application.

**2. Execution**

This involves executing these tests plans, measuring, collecting data and verifying if it meets

the quality criteria set in the quality plan. Data collected is used to make appropriate changes

in the plan related to development and testing. We have used the prepared documentation for

the testing of Application.

* **Functional testing:-**

The next level of testing is the functional testing which consist of integrated and system testing .The integration testing many tested modules is combined into sub-systems, which are then tested. The goal there is to see if the modules can be integrated properly, the emphasis being on testing interfaces between modules. This activity can be considered as testing the design, and hence the emphasis on testing module interactions and in system testing the entire software system is tested. The reference document for this process is requirement document, and the goal is to see if the software meets its requirements. This is essentially a validation exercise.

* **Structural testing:**

The first level of testing is structural testing which is also known as unit testing. In this different modules are tested against the specifications produced during design for the modules. Structural testing is essential for verification of the code produced during the coding phase and hence the goal is to test the internal logic of the modules.

* **Levels of Testing**

In order to uncover the errors, present in different phases, we have the concept of levels of

testing. The basic levels of testing are:

Needs Acceptance Testing

Requirements System Testing

Design Testing Integration Testing

Code Unit Testing

* **Unit Testing**

This focus on verification effort on smallest unit of Shopping Mall Website designed the software component or module. In this we can take User Authentication module is considered to be as unit testing part. Using component level design, description as a guide important control path is tested to uncover errors. The module interface is tested to ensure that information properly flows into and out of the program unit under test. Data structure is locally examined to ensure that data stored temporarily maintains its integrity during all the steps in execution of algorithm. Boundary conditions are tested to ensure that module is operating properly at boundaries which are established to limit or restrict processing. All independent paths through control structure are exercised to ensure that all statements in module are executed at least once. Finally all error-handling paths are tested. In unit testing of shopping mall website we tested each and every page of website individually. If error comes then we corrected that error and then we proceeded to next step i.e. testing of next page.

* **Integration Testing**

In this process of testing it is incremented approach to construction of program structure. Modules are integrated moving downward beginning with main control module. Modules subordinate structure to main control module is incorporated into structure. This form of testing is performed of software in five steps: -.

Main module “Purchasing of products” is tested after integrating and linking the

pages of online purchasing and online payment.

After testing of main module we proceed to other modules one by one. Other modules

are seller tasks in mall and the tasks of admin inside mall.

Tests are conducted as each component is integrated.

On completing each set of tests another stub is replaced.

* **Top-Down Integration Testing**

In well-factored program structure decision-making occurs at upper levels in hierarchy and

therefore, encountered first. If major control problem do exist, early recognition is essential.

This is termed as.

* **Bottom-up integration testing**

It begins construction and testing with atomic modules as the components are integrated from

the bottom-up, processing required for components subordinate to a given level is always

available and the need for stubs is eliminated.

* **System Testing**

Here the entire software system is tested. The reference document for this process is the requirements document, and the goal is to see if software meets its requirements. Here entire Software has been tested against requirements of project and it is checked whether all requirements of project have been satisfied or not. At last when unit testing and integration testing is completed then we go for system testing i.e. after combining all modules of the website of mall we test it for the proper functioning and proper dataflow between different modules of the project. If the dataflow and the operations between different modules are working correctly then product is ready to deliver to customer.

* **Acceptance Testing**

Acceptance Testing is performed with realistic data of the client to demonstrate that the software is working satisfactorily. Testing here is focused on external behavior of the system; the internal logic of program is not emphasized. Test cases should be selected so that the largest number of attributes of an equivalence class is exercised at once. The testing phase is an important part of software development. It is the process of finding errors and missing operations and also a complete verification to determine whether the objectives are met and the user requirements are satisfied. The client tests the website of hotel to check the requirement given to the software developer’s team and the working of the actual project. If the website fulfills the requirements of the client the it satisfies the acceptance testing and if client find something wrong in the product then he can ask the developers to correct that problem in project.

**9.1 Testing the project:**

This application was tested on Android Phone Emulator.

When running an Android Phone application for the first time in Android Phone

Emulator, the following Steps should be performed then these events will occur

* Open Android Virtual Device Manager
* Add new AVD and Set the Name, Device, and Target of that device.
* Start the device
* The emulator starts.
* The emulator loads the Android Phone operating system.
* The emulator displays the Android Phone Start screen.
* The application is deployed to the emulator.
* Then application runs on the emulator
* For this application above step were used to test the code in emulator and if error

occurs then correction was done.

* Another way of testing was through installing apk file in android phone. The

following steps were used to run application in phone

* Run the application in emulator then seed the apk file in mobile from projects bin

folder

* Install the application in mobile.
* It will provide the actual installation of application in phone.

**9.1.1Black Box Testing**

It is designed to uncover errors. They are used to demonstrate that website function are operations; that input is properly accepted and output is correctly produced; and that integrity of external information is maintained. A black box examines some fundamental aspects of a system with little regard for the internal logical structure of the software.

**9.1.2 White Box Testing**

It is predicated on close examination of procedural details. Providing test cases that exercise specific set of conditions and loops test logical paths through the software. The “state of the program” may be examined at various points to determine if the expected or asserted status corresponds to the actual status.

In my project I have executed acceptance testing, because I have implemented all the

**Chapter 10: Implementation**

Implementation is the process of having system personally checked out and put new equipment’s into use, trained users, installed the new application and construct and file of data needed to use it. Depending on the size of the organization that will be involved in using the application and the risk associated with its users, system developers may choose to pilot (test) the operation in only one area of the firm, say in one department or with only one or two persons. Regardless of the implementation strategy use, developers strive to ensure that the system’s initial use in trouble. Once installed, applications are often used for many years. However, both the organization and the user will change, and the environment will be different over weeks and months. Therefore, the application will undoubtedly have to be maintained, modifications and change will be made to software’s, file or procedures to meet emerging user requirement. In the sense, Implementation is an ongoing process

Implementation is the stage of a project during which the design of a system is tested, debugged and made operational. Implementation is the process of having systems personnel check on output, new equipment into use, train users, install the new equipment and construct any files of data needed to use it. Once installed application are often used for many years. However, both the organization and users will change, and the environment will be different over weeks and months. Therefore, the application will undoubtedly have to be maintained; modification and changes will be made to the software, files or producers to meet user requirements. During the time of implementation of this software file setup starts, gathered all the documentation, user manuals, procedures manuals, computer operating instruction and security procedures.

Following activities will be performed for implementation:

* Study Requirements and Design document
* Decide methodology of development
* Select and customize coding standards
* Develop units
* Code
* Self-test will be done before submitting it to Code review
* Code review will be done by peers, by the method of walkthrough
* Unit test by test engineers.
* Integration of components
* Integration testing
* Preparation of Installable
* Release to testing team
* Outputs
* Integrated and tested units
* Peer Review
* Code Review Log
* Defect List

**Post Implementation and Software maintenance**

Software Maintenance is the last part of the System Development Life Cycle which is actually the implementation of the post-implementation review plan. Software maintenance is a very broad activity that includes error corrections, enhancements of capabilities, deletion of obsolete capabilities, and optimization. Because change is inevitable, mechanism must be developed for evaluation, controlling and making modifications. There are four types of maintenance:

• Corrective Maintenance

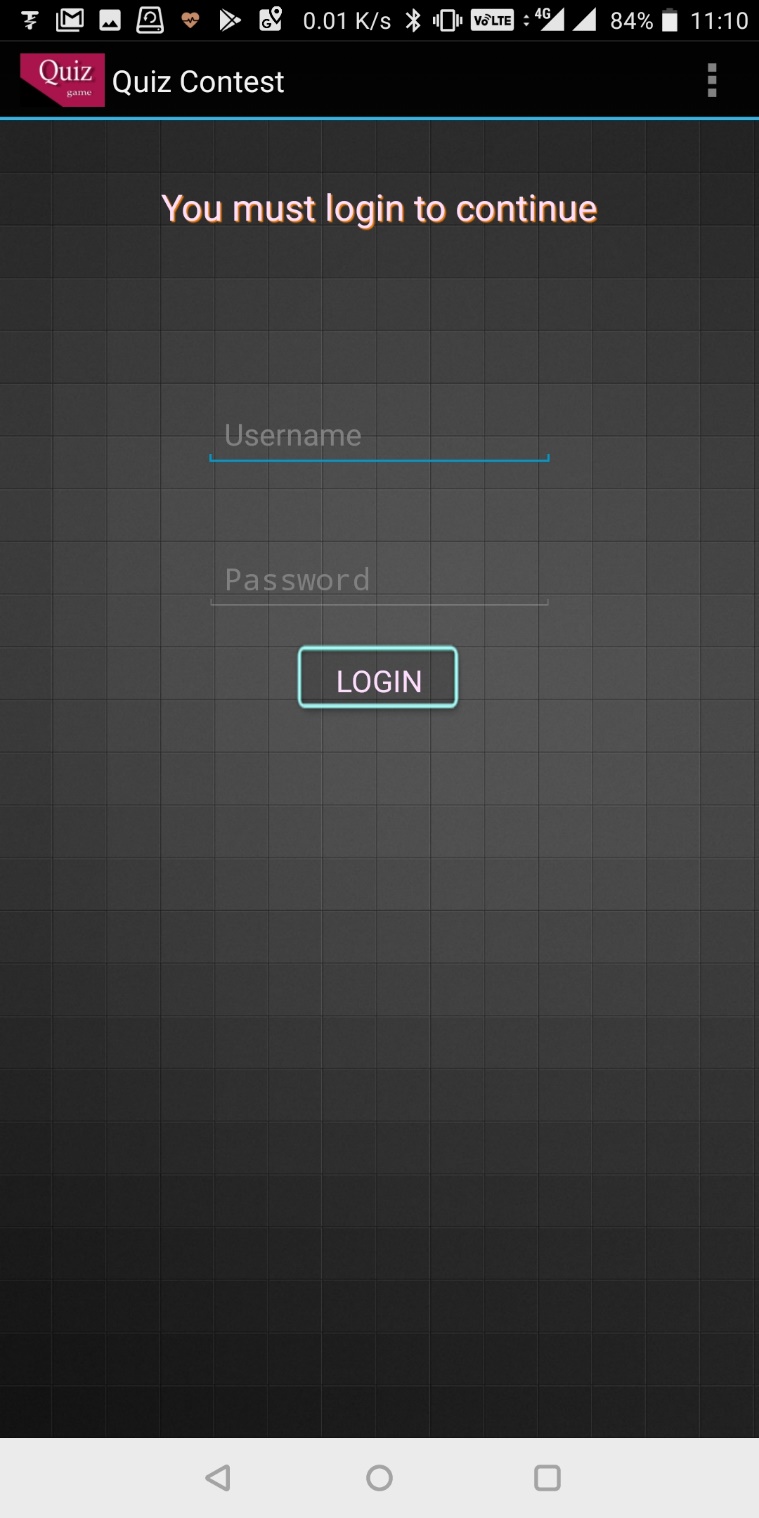
• Adaptive Maintenance

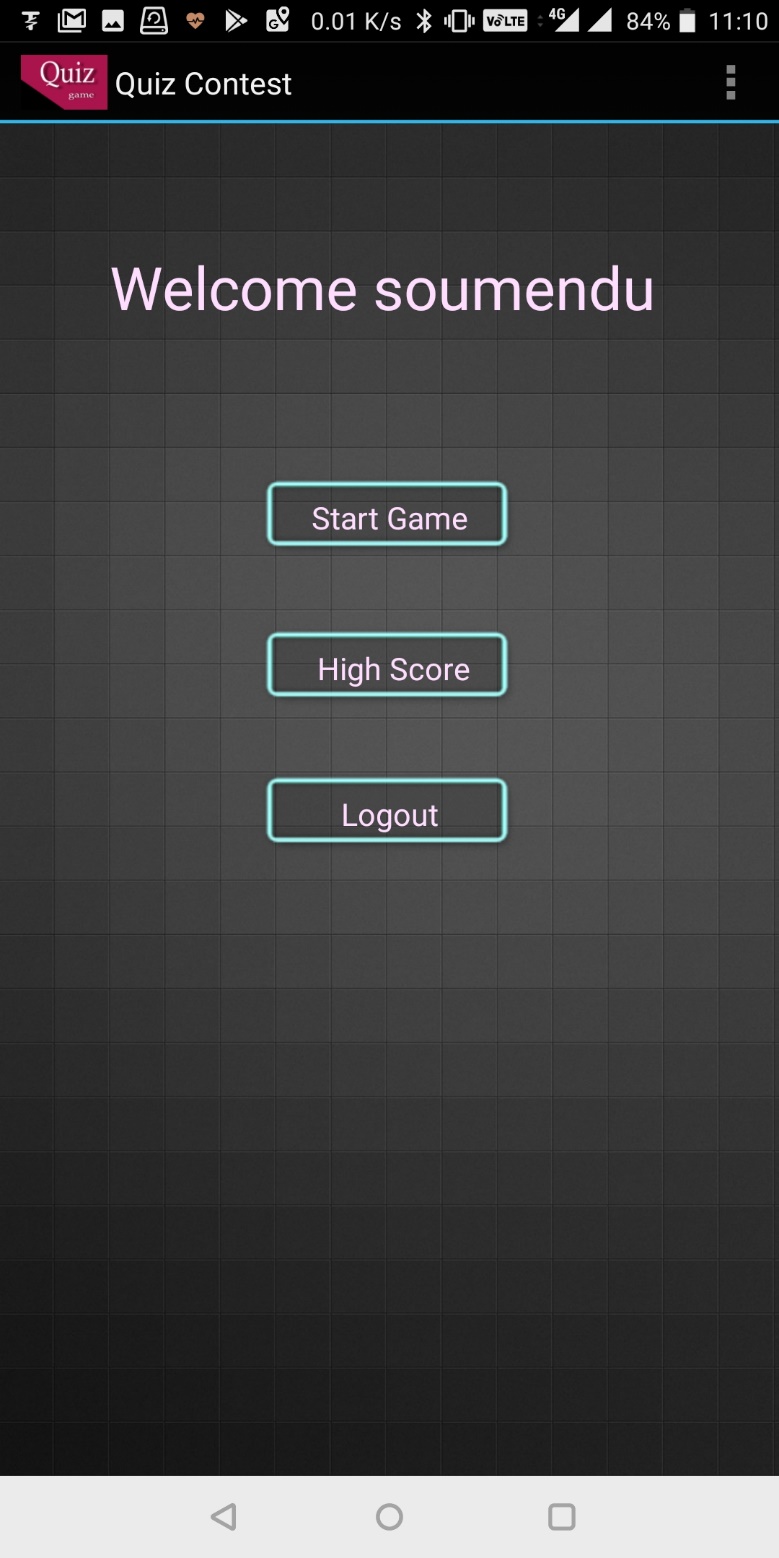
• Perfective Maintenance

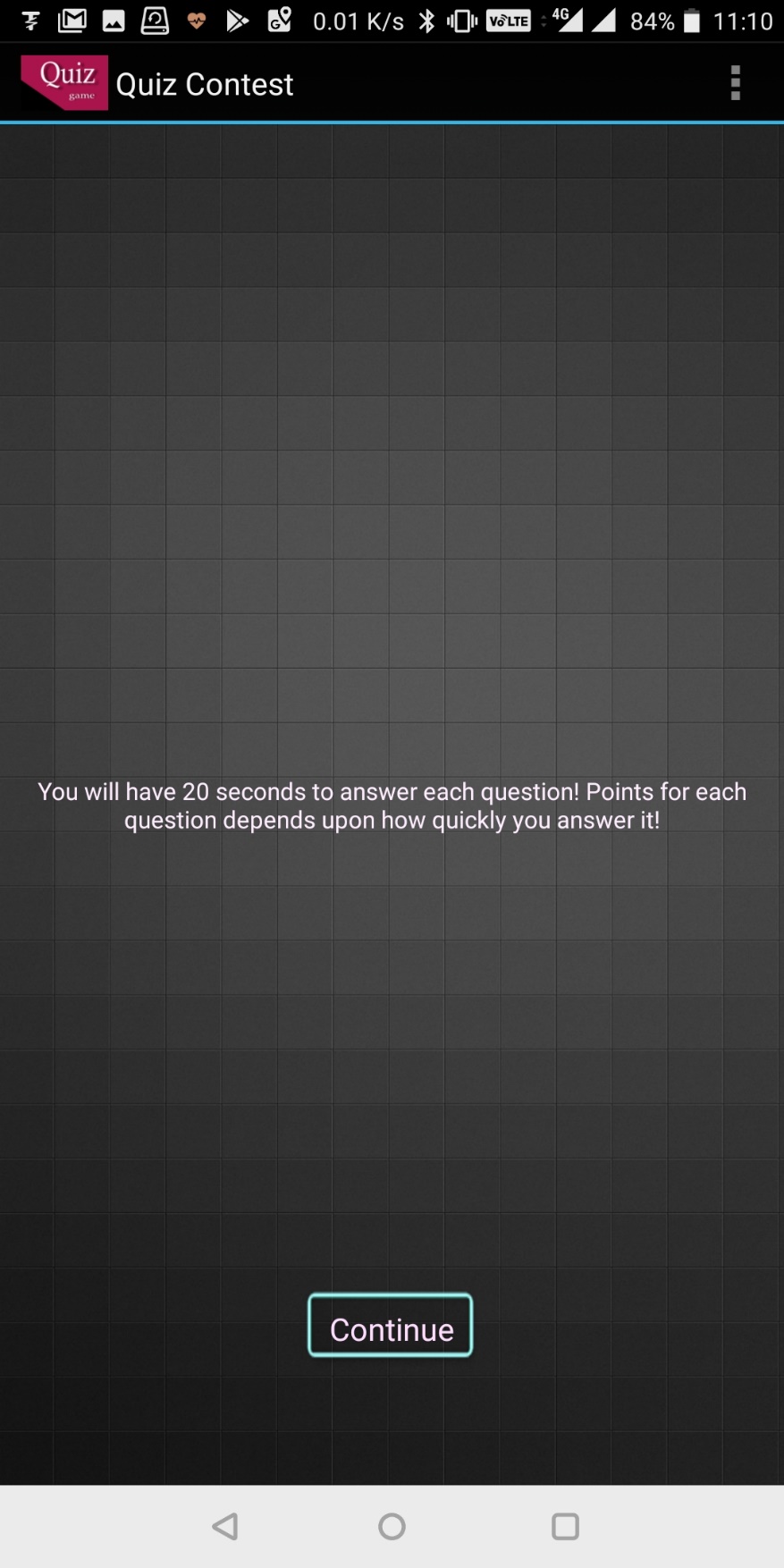
• Preventive Maintenance

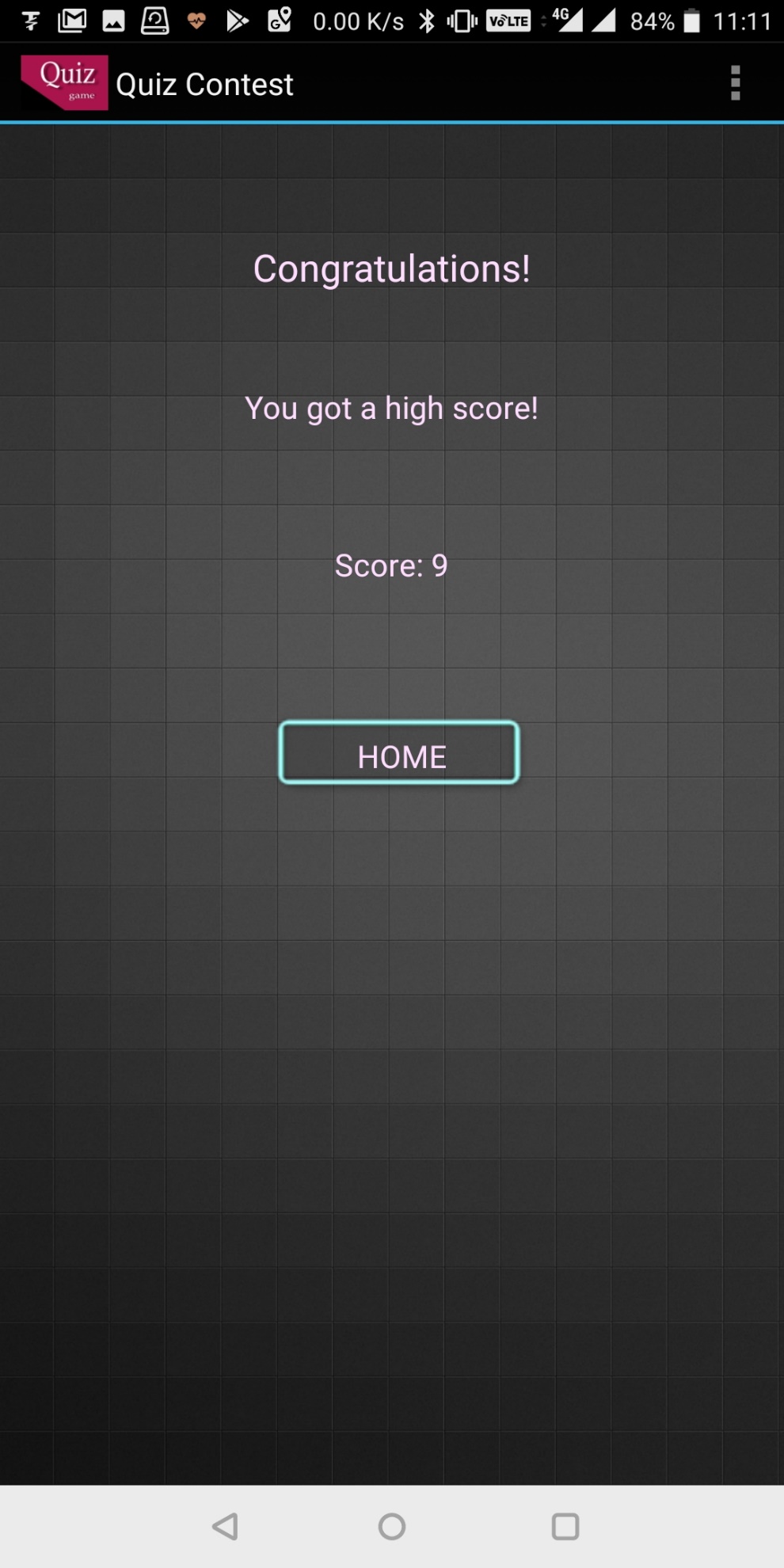
When this system is installed it is used for long period. However, this period of use brings with it the need to continually maintain the system, but this system can be modified and new technologies can be used which are prevalent in market at that period of time. Software maintenance requires an accurate maintenance plan to be prepared during the software development. It should specify how users will request modifications or report problems. The budget should include resource and cost estimates. A new decision should be addressed for the developing of every new system feature and its quality objectives. Software maintenance takes more effort than all other phases of software life cycle, but it has not been given as much importance as it deserved. It is an admitted fact that approximately 60 to 70% effort is spent on maintenance phase of software development life cycle.

**Chapter 11: Screen Layer**

****

****

****

****

**Chapter 12: Conclusion**

The ONLINE QUIZ is an android application for to take online test in an efficient manner and no time wasting for checking the paper. The main objective of ONLINE QUIZ is to efficiently evaluate the candidate thoroughly through a fully automated system that not only saves lot of time but also gives fast results. For students they give papers according to their convenience and time and there is no need of using extra thing like paper, pen etc. This can be used in educational institutions as well as in corporate world. Can be used anywhere any time as it is an android based application (user location doesn’t matter). No restriction that examiner has to be present when the candidate takes the test.

.

**REFERENCES**

1. Corona Documentation retrieved from corona developers

[file:///P:/CoronaApiDocs/index.html](file:///P:\CoronaApiDocs\index.html)

1. Er.Vogella

<http://www.vogella.com/>

1. Book of java

“Head First Java”

1. Experts Opinion

Stack overflow and interview street

1. Other Websites:

http://www.openhandsetalliance.com

Blog <http://android-developers.blogspot.com>

<http://www.anddev.org>

<http://www.wikipedia.com/androidhistory>

<http://developers.android.com>